AI VIETNAM
All-in-One Course

Vectorized Implementation for Linear Regression

Quang-Vinh Dinh
Ph.D. in Computer Science

Year 2023

Outline

- Review on One-sample Training
- Vectorize the Linear Regression (m-sample)
- Vectorize the Linear Regression (N-sample)
- Proofs of Some Matrix Properties

Review

Transpose

$$\vec{v} = \begin{bmatrix} v_1 \\ \dots \\ v_n \end{bmatrix} \qquad \vec{v}^T = [v_1 \quad \dots v_n]$$

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}^T \rightarrow \begin{bmatrix} 1 & 2 \end{bmatrix}$$

$$\mathrm{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \qquad \mathrm{A}^T = \begin{bmatrix} a_{11} & \dots & a_{m1} \\ \dots & \dots & \dots \\ a_{1n} & \dots & a_{mn} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T \rightarrow \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

```python
import numpy as np

# create data
data = np.array([1,2,3])
factor = 2

# broadcasting
result_multiplication = data*factor
```

```
[1 2 3]
[2 4 6]
```

Multiply with a number

$$\alpha\vec{u} = \alpha \begin{bmatrix} u_1 \\ \dots \\ u_n \end{bmatrix} = \begin{bmatrix} \alpha u_1 \\ \dots \\ \alpha u_n \end{bmatrix}$$

$$\underset{\text{data}}{\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}} * \begin{bmatrix} 2 \end{bmatrix} = \underset{\text{result}}{\begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}}$$

Review

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7
x	y

Traditional

1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

$$\hat{y} = wx + b \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} \rightarrow \boldsymbol{\theta}^T = [b \quad w]$$

$$\hat{y} = wx + b1 = [b \quad w]\begin{bmatrix} 1 \\ x \end{bmatrix} = \boldsymbol{\theta}^T \boldsymbol{x}$$

dot product

Review

1) Pick a sample (x, y) from training data

Traditional

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

$$\hat{y} = wx + b \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x}$$

$$L(\hat{y}, y) = (\hat{y} - y)^2$$

numbers

What will we do?

Review

1) Pick a sample (x, y) from training data

Traditional

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

$$\hat{y} = wx + b \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\begin{cases} \dfrac{\partial L}{\partial b} = 2(\hat{y} - y) = 2 \times (\hat{y} - y) \times 1 \\ \dfrac{\partial L}{\partial w} = 2x(\hat{y} - y) = 2 \times (\hat{y} - y) \times x \end{cases}$$

$$\begin{bmatrix} 2 \times (\hat{y} - y) \times 1 \\ \underbrace{2 \times (\hat{y} - y)} \times x \end{bmatrix} = \underline{2(\hat{y} - y)} \begin{bmatrix} 1 \\ x \end{bmatrix} = 2(\hat{y} - y)\boldsymbol{x} = \begin{bmatrix} \dfrac{\partial L}{\partial b} \\ \dfrac{\partial L}{\partial w} \end{bmatrix} = L'_{\boldsymbol{\theta}} \qquad \rightarrow \qquad L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y)$$

common factor

Review

1) Pick a sample (x, y) from training data

Traditional

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

$$\hat{y} = wx + b$$

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$L'_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{bmatrix}$$

$$\left\{ \begin{array}{l} \boxed{\begin{matrix} b \\ w \\ \boldsymbol{\theta} \end{matrix}} = \boxed{\begin{matrix} b \\ w \\ \boldsymbol{\theta} \end{matrix}} - \eta \boxed{\begin{matrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \\ L'_{\boldsymbol{\theta}} \end{matrix}} \end{array} \right.$$

$$\rightarrow \quad \boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

Review

1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

Traditional

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate

Vectorized

$$\hat{y} = wx + b$$

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$$

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7
x	y

1 $\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ 6.7 \end{bmatrix}$

Given $\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$

$\eta = 0.01$

1) Pick a sample $(\mathbf{x}, y)$ from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

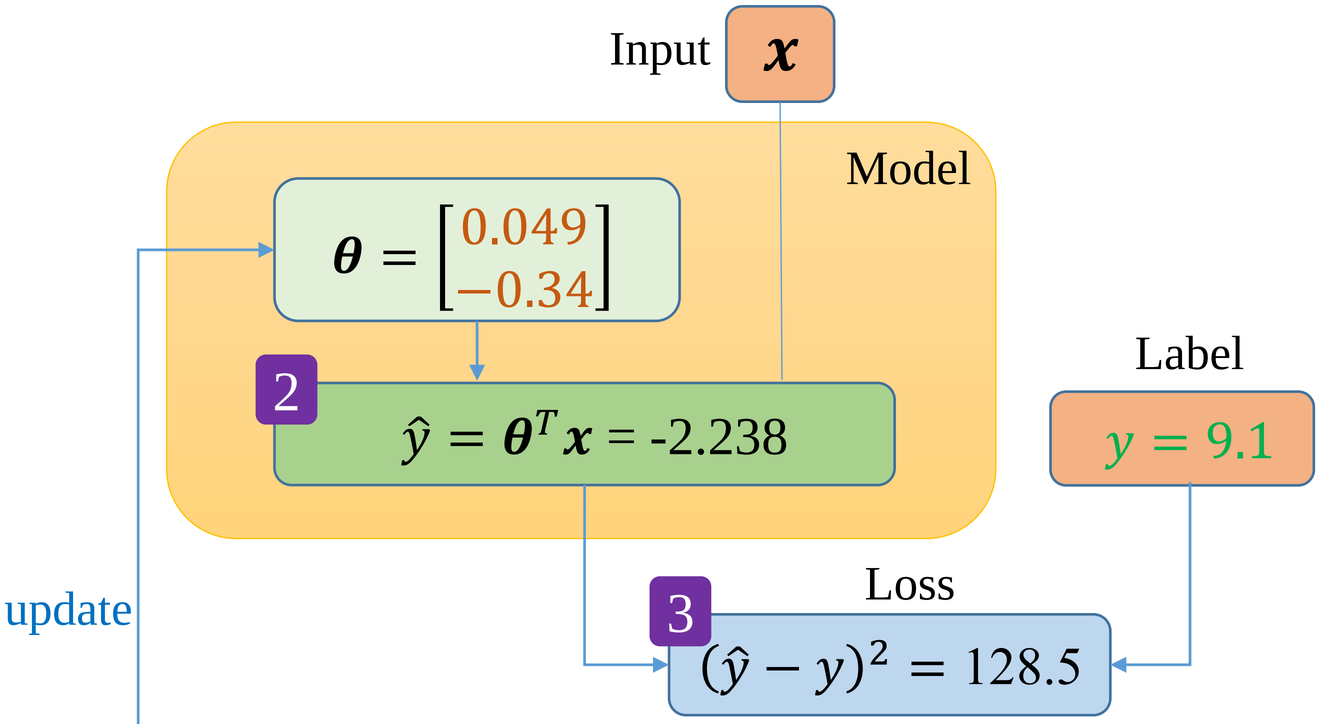
4) Compute derivative

$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate



4 $L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y) = \begin{bmatrix} -22.658 \\ -151.81 \end{bmatrix}$

5 $\boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix} - \eta \begin{bmatrix} -22.658 \\ -151.81 \end{bmatrix} = \begin{bmatrix} 0.2755 \\ 1.1781 \end{bmatrix}$

Objective

❖ Implementation (vectorization using numpy)

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate

```python
import numpy as np

# forward
def predict(x, theta):
    return x.dot(theta)

# compute gradient
def gradient(y_hat, y, x):
    dtheta = 2*x*(y_hat-y)

    return dtheta

# update weights
def update_weight(theta, lr, dtheta):
    dtheta_new = theta - lr*dtheta

    return dtheta_new
```

$\hat{y} = wx + b$

$\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} \quad \boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix}$

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7
x	y

1 $\boldsymbol{x} = \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ 3.5 \end{bmatrix}$

Given $\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix}$

$\eta = 0.01$

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

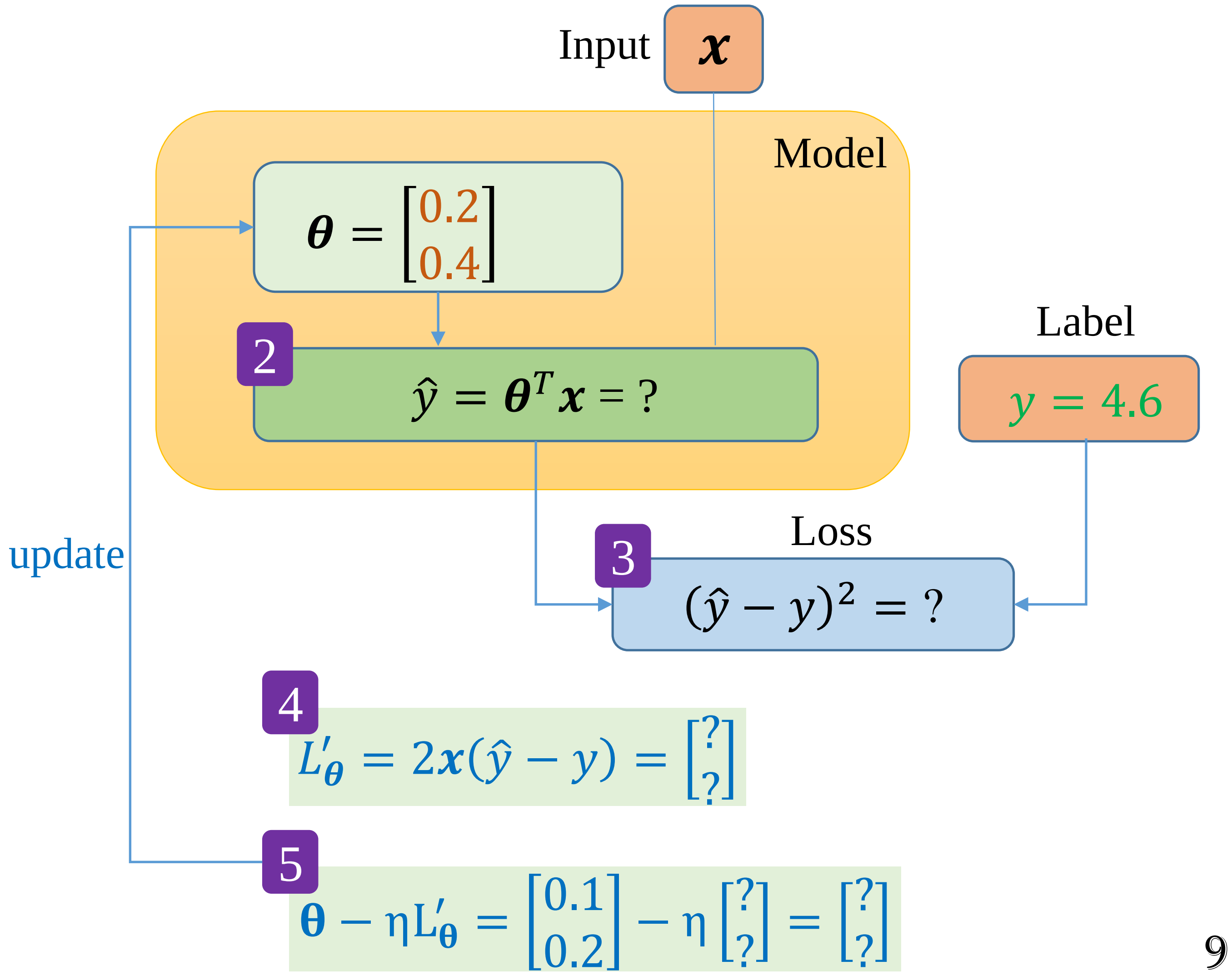
4) Compute derivative

$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate



```python
data = np.genfromtxt('data.csv', delimiter=',')
N = 4

areas  = data[:, 0].reshape(N, 1)
prices = data[:, 1].reshape(N,)

# vector [1, area]
features = np.hstack([np.ones((N,1)), areas])
```

```python
# forward
def predict(x, theta):
    return x.T.dot(theta)

# compute gradient
def gradient(y_hat, y, x):
    dtheta = 2*x*(y_hat-y)
    return dtheta

# update weights
def update_weight(theta, lr, dtheta):
    dtheta_new = theta - lr*dtheta
    return dtheta_new
```

```python
lr = 0.01
epoch_max = 10

# [b, w]
theta = np.array([0.049, -0.34])

for epoch in range(epoch_max):
    for i in range(N):
        # get a sample
        x = features[i,:]
        y = prices[i]

        # predict y_hat
        y_hat = predict(x, theta)

        # compute loss
        loss = (y_hat-y)*(y_hat-y)

        # compute gradient
        dtheta = gradient(y_hat, y, x)

        # update weights
        theta = update_weight(theta, lr, dtheta)
```

Advertising Problem

$$\hat{y} = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + b$$

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \theta = \begin{bmatrix} b \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

Features			Label
TV	**Radio**	**Newspaper**	**Sales**
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9

Advertising data

if TV=55.0, Radio=34.0, and Newspaper=62.0, price=?

1) Pick a sample $(\mathbf{x}, y)$ from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate

Outline

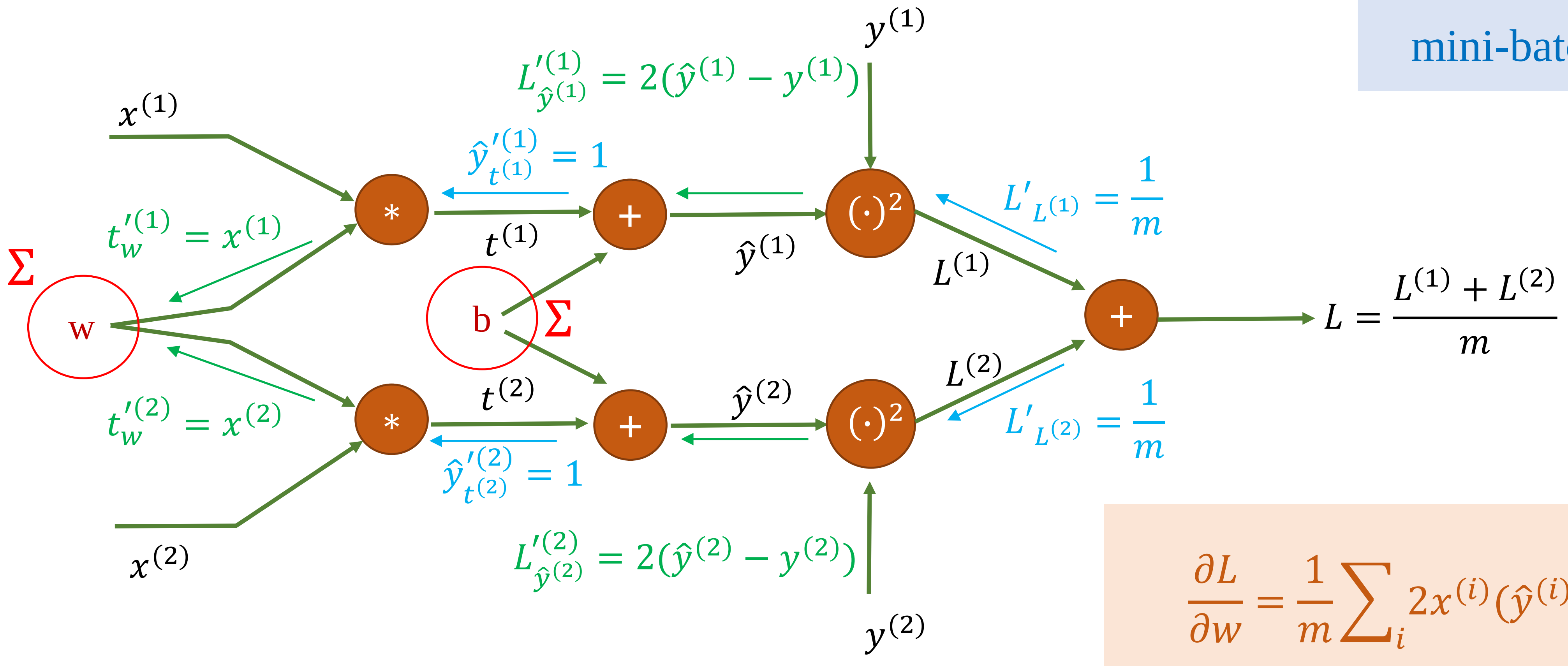
- Review on One-sample Training
- Vectorize the Linear Regression (m-sample)
- Vectorize the Linear Regression (N-sample)
- Proofs of Some Matrix Properties

Linear Regression (m-samples)

	Feature	Label
	area	price
	6.7	9.1
	4.6	5.9

❖ Compute derivate for w and b

mini-batch m = 2



$$\frac{\partial L}{\partial w} = \frac{1}{m}\sum_i 2x^{(i)}(\hat{y}^{(i)} - y^{(i)})$$

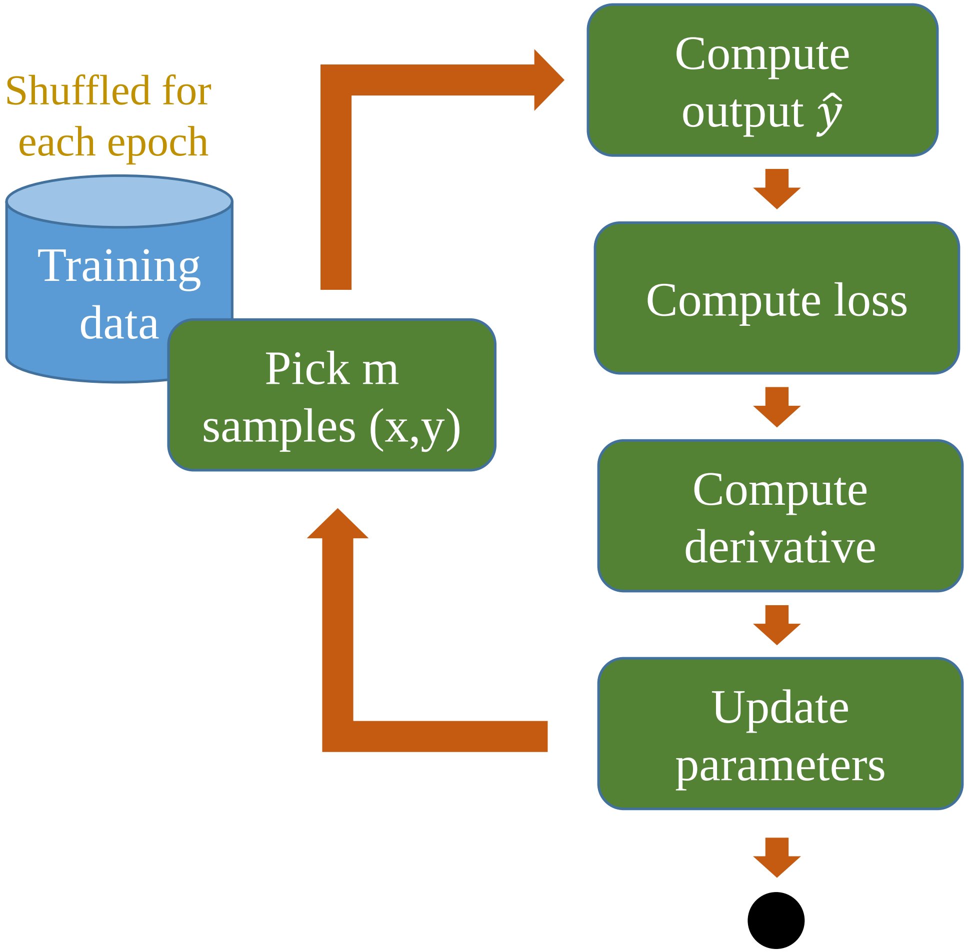
$$\frac{\partial L}{\partial b} = \frac{1}{m}\sum_i 2(\hat{y}^{(i)} - y^{(i)})$$

Linear Regression (m-samples)

Feature	Label
area	price
6.7	9.1
4.6	5.9

❖ House price prediction

❖ m-sample training (1<m<N)



1) Pick m samples $(x^{(i)}, y^{(i)})$ from training data

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \leq i < m$$

3) Compute loss

$$L^{(i)} = (\hat{y}^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < m$$

4) Compute derivative

$$L_w^{\prime(i)} = 2x^{(i)}(\hat{y}^{(i)} - y^{(i)})$$
$$L_b^{\prime(i)} = 2(\hat{y}^{(i)} - y^{(i)}) \quad \text{for } 0 \leq i < m$$

5) Update parameters

$$w = w - \eta \frac{\sum_i L_w^{\prime(i)}}{m}$$

$$b = b - \eta \frac{\sum_i L_b^{\prime(i)}}{m}$$

Learning rate η

Linear Regression (m-samples)

Feature	Label
area	price
6.7	9.1
4.6	5.9

1) Pick a sample $(\mathbf{x}, y)$ from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$L'_{\boldsymbol{\theta}} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

η is learning rate

1) Pick m samples $(\mathbf{x}^{(i)}, y^{(i)})$ from training data

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \le i < m$$

3) Compute loss

$$L^{(i)} = (\hat{y}^{(i)} - y^{(i)})^2 \quad \text{for } 0 \le i < m$$

4) Compute derivative

$$L'^{(i)}_{\boldsymbol{\theta}} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \text{ for } 0 \le i < m$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L'^{(i)}_{\boldsymbol{\theta}}}{m}$$

η is learning rate

Linear Regression (m-samples)

1) Pick m samples $(\mathbf{x}^{(i)}, y^{(i)})$ from training data

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} = \left(\boldsymbol{x}^{(i)}\right)^T \boldsymbol{\theta} \quad \text{for } 0 \leq i < m$$

3) Compute loss

$$L^{(i)} = (\hat{y}^{(i)} - y^{(i)})^2 \quad \text{for } 0 \leq i < m$$

4) Compute derivative

$$L'^{(i)}_{\boldsymbol{\theta}} = 2\boldsymbol{x}^{(i)}(\hat{y}^{(i)} - y^{(i)}) \text{ for } 0 \leq i < m$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{\sum_i L'^{(i)}_{\boldsymbol{\theta}}}{m}$$

η is learning rate

```python
for epoch in range(epoch_max):
    for j in range(0, data_size, m):

        # some variables
        sum_of_losses = 0
        gradients = np.zeros((2,))
        for index in range(j, j+m):
            # get mini-batch
            x_i = data[index]
            y_i = prices[index]

            # predict y_hat_i
            y_hat_i = x_i.dot(theta)

            # compute loss
            l_i = (y_hat_i - y_i)*(y_hat_i - y_i)

            # compute gradient
            gradient_i = x_i*2*(y_hat_i - y_i)

            # accumulate gradients
            gradients = gradients + gradient_i
            sum_of_losses = sum_of_losses + l_i

        # normalize
        sum_of_losses = sum_of_losses/2
        gradients     = gradients/2
```

Linear Regression (m-samples)

❖ Construct formulas

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

2) Compute output $\hat{y}$

$$\hat{y} = wx + b$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix} = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \end{bmatrix} = \begin{bmatrix} wx^{(1)} + b \\ wx^{(2)} + b \end{bmatrix} = \begin{bmatrix} 1 & x^{(1)} \\ 1 & x^{(2)} \end{bmatrix} \begin{bmatrix} b \\ w \end{bmatrix} = \boldsymbol{x\theta} = \begin{bmatrix} -2.229 \\ -1.515 \end{bmatrix}$$

Linear Regression (m-samples)

❖ Construct formulas

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

3) Compute loss

$L = (\hat{y} - y)^2$

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \end{bmatrix} = \boldsymbol{x\theta}$$

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

$$\boldsymbol{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \begin{bmatrix} L\left(\hat{y}^{(1)}, y^{(1)}\right) \\ L\left(\hat{y}^{(2)}, y^{(2)}\right) \end{bmatrix} = \begin{bmatrix} (\hat{y}^{(1)} - y^{(1)})^2 \\ (\hat{y}^{(2)} - y^{(2)})^2 \end{bmatrix} = \begin{bmatrix} \left(\hat{y}^{(1)} - y^{(1)}\right) * \left(\hat{y}^{(1)} - y^{(1)}\right) \\ \left(\hat{y}^{(2)} - y^{(2)}\right) * \left(\hat{y}^{(2)} - y^{(2)}\right) \end{bmatrix}$$

$$= \begin{bmatrix} \left(\hat{y}^{(1)} - y^{(1)}\right) \\ \left(\hat{y}^{(2)} - y^{(2)}\right) \end{bmatrix} \odot \begin{bmatrix} \left(\hat{y}^{(1)} - y^{(1)}\right) \\ \left(\hat{y}^{(2)} - y^{(2)}\right) \end{bmatrix} = (\hat{\boldsymbol{y}} - \boldsymbol{y}) \odot (\hat{\boldsymbol{y}} - \boldsymbol{y})$$

Hadamard product

AI VIETNAM
All-in-One Course

Linear Regression (m-samples)

❖ Construct formulas

	Feature	Label
	area	price
	6.7	9.1
	4.6	5.9
	3.5	4.6
	5.5	6.7

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

4) Compute derivative

$$\frac{\partial L}{\partial b} = 2(\hat{y} - y) \qquad \frac{\partial L}{\partial w} = 2x(\hat{y} - y)$$

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \end{bmatrix} = \boldsymbol{x\theta}$$

one sample

$$L'_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{bmatrix} = 2(\hat{y} - y)\begin{bmatrix} 1 \\ x \end{bmatrix} = 2(\hat{y} - y)\boldsymbol{x}$$

two samples

$$2(\hat{\boldsymbol{y}} - \boldsymbol{y}) = \begin{bmatrix} 2(\hat{y}^{(1)} - y^{(1)}) \\ 2(\hat{y}^{(2)} - y^{(2)}) \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

Linear Regression (m-samples)

❖ Construct formulas

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

one sample

$$L'_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{bmatrix} = 2(\hat{y} - y)\begin{bmatrix} 1 \\ x \end{bmatrix} = 2(\hat{y} - y)\boldsymbol{x}$$

propagate and accumulate

two samples

$$k = 2(\hat{\boldsymbol{y}} - \boldsymbol{y}) = \begin{bmatrix} 2(\hat{y}^{(1)} - y^{(1)}) \\ 2(\hat{y}^{(2)} - y^{(2)}) \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

how?

AI VIETNAM
All-in-One Course

Linear Regression (m-samples)

❖ Construct formulas

propagate and accumulate

	Feature	Label
	area	price
	6.7	9.1
	4.6	5.9
	3.5	4.6
	5.5	6.7

two samples

$$k = 2(\hat{\boldsymbol{y}} - \boldsymbol{y}) = \begin{bmatrix} 2(\hat{y}^{(1)} - y^{(1)}) \\ 2(\hat{y}^{(2)} - y^{(2)}) \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

way 1

$$\begin{bmatrix} 2(\hat{y}^{(1)} - y^{(1)}) & 2(\hat{y}^{(2)} - y^{(2)}) \end{bmatrix} \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial b} & \frac{\partial L}{\partial w} \end{bmatrix} = \boldsymbol{k}^T \boldsymbol{x}$$

way 2

$$\begin{bmatrix} 1 & 1 \\ 6.7 & 4.6 \end{bmatrix} \begin{bmatrix} 2(\hat{y}^{(1)} - y^{(1)}) \\ 2(\hat{y}^{(2)} - y^{(2)}) \end{bmatrix} = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{bmatrix} = \boldsymbol{x}^T \boldsymbol{k}$$

Linear Regression (m-samples)

$$L'_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{bmatrix} = \boldsymbol{x}^T \boldsymbol{k}$$

❖ Construct formulas

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

$$L'_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial L}{\partial b} & \frac{\partial L}{\partial w} \end{bmatrix} = \boldsymbol{k}^T \boldsymbol{x}$$

$$\begin{cases} b = b - \eta \frac{\partial L}{\partial b} \\ w = w - \eta \frac{\partial L}{\partial w} \end{cases}$$

$$\boldsymbol{\theta} \quad \boldsymbol{\theta} \quad L'_{\boldsymbol{\theta}}$$

which one?

$$\rightarrow \quad \boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

Linear Regression (m-samples)

❖ Example

$\eta = 0.01$

$m = 2$

	Feature	Label	
	area	price	
	6.7	9.1	
	4.6	5.9	
	3.5	4.6	
	5.5	6.7	

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

$$\hat{y} = \boldsymbol{x\theta} = \begin{bmatrix} -2.229 \\ -1.515 \end{bmatrix}$$

$$\boldsymbol{L} = (\hat{\boldsymbol{y}} - \boldsymbol{y}) \odot (\hat{\boldsymbol{y}} - \boldsymbol{y}) = \begin{bmatrix} 128.3 \\ 54.98 \end{bmatrix}$$

$$\boldsymbol{k} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y}) = \begin{bmatrix} -22.658 \\ -14.830 \end{bmatrix}$$

$$L'_{\boldsymbol{\theta}} = \boldsymbol{x}^T \boldsymbol{k} = \begin{bmatrix} 1 & 1 \\ 6.7 & 4.6 \end{bmatrix} \begin{bmatrix} -22.658 \\ -14.830 \end{bmatrix} = \begin{bmatrix} -37.488 \\ -220.02 \end{bmatrix}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{L'_{\boldsymbol{\theta}}}{m}$$

$$= \begin{bmatrix} -0.34 \\ 0.049 \end{bmatrix} - \frac{0.01}{2} \begin{bmatrix} -37.488 \\ -220.02 \end{bmatrix} = \begin{bmatrix} 0.236 \\ 0.760 \end{bmatrix}$$

Linear Regression (m-samples)

❖ Formulas

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \end{bmatrix}$$

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$

1) Pick m samples $(\boldsymbol{x}, \boldsymbol{y})$ from training data

2) Compute output $\hat{y}$

$$\hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{\theta}$$

3) Compute loss

$$\boldsymbol{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = (\hat{\boldsymbol{y}} - \boldsymbol{y}) \odot (\hat{\boldsymbol{y}} - \boldsymbol{y})$$

4) Compute derivative

$$\boldsymbol{k} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \boldsymbol{x}^T \boldsymbol{k}$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{L'_{\boldsymbol{\theta}}}{m}$$

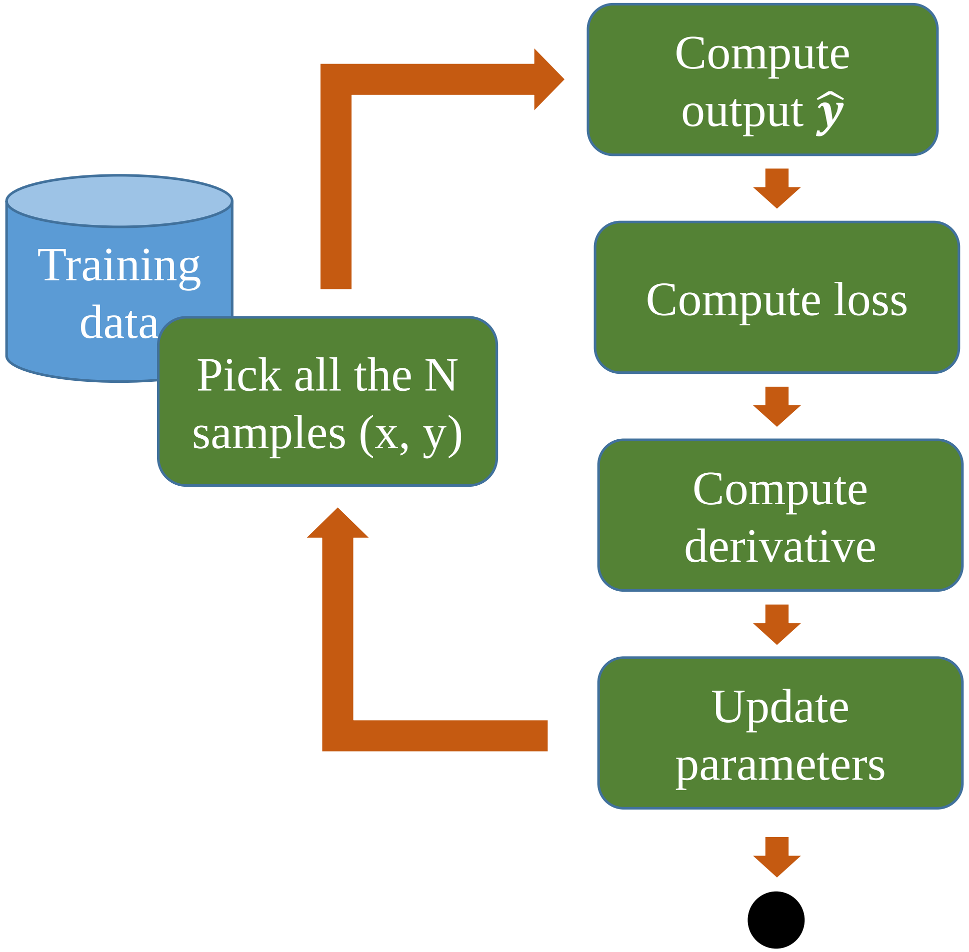
Outline

- Review on One-sample Training
- Vectorize the Linear Regression (m-sample)
- Vectorize the Linear Regression (N-sample)
- Proofs of Some Matrix Properties

Linear Regression (m-samples)

❖ House price prediction

❖ N-sample training



1) Pick all the N samples $(x^{(i)}, y^{(i)})$ from training data

2) Compute output $\hat{y}^{(i)}$

$$\hat{y}^{(i)} = wx^{(i)} + b \qquad \text{for } 0 \le i < N$$

3) Compute loss

$$L^{(i)} = (\hat{y}^{(i)} - y^{(i)})^2 \quad \text{for } 0 \le i < N$$

4) Compute derivative

$$L_w'^{(i)} = 2x^{(i)}(\hat{y}^{(i)} - y^{(i)})$$
$$L_b'^{(i)} = 2(\hat{y}^{(i)} - y^{(i)}) \qquad \text{for } 0 \le i < N$$

5) Update parameters

$$w = w - \eta \frac{\sum_i L_w'^{(i)}}{N}$$

$$b = b - \eta \frac{\sum_i L_b'^{(i)}}{N} \qquad \text{Learning rate } \eta$$

Linear Regression

Feature	Label
area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

House Price Data

$$\boldsymbol{x} = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \\ 1 & 3.5 \\ 1 & 5.5 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

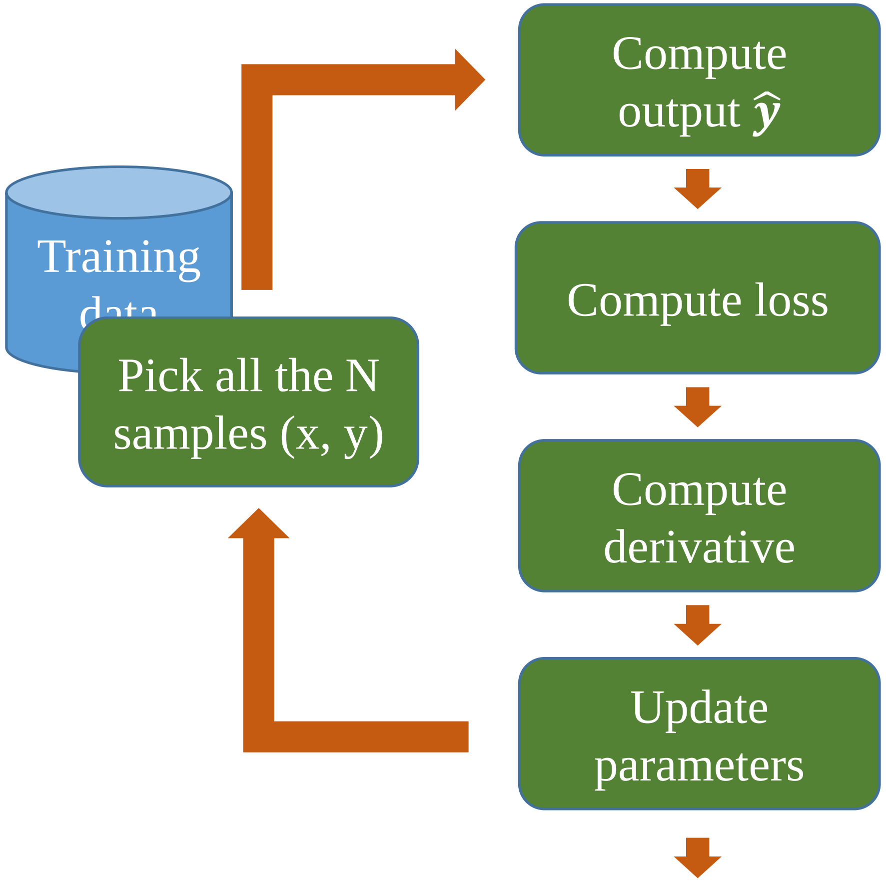
Model

$$\text{price} = \text{w} * area + b$$

$$\hat{y} = wx + b$$

Parameter Initialization

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$



1) Pick all the N samples from training data

2) Compute output $\hat{\mathbf{y}}$

$$\hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{\theta}$$

3) Compute loss

$$\boldsymbol{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = (\hat{\boldsymbol{y}} - \boldsymbol{y}) \odot (\hat{\boldsymbol{y}} - \boldsymbol{y})$$

4) Compute derivative

$$\boldsymbol{k} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \boldsymbol{x}^T \boldsymbol{k}$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{L'_{\boldsymbol{\theta}}}{N}$$

η is learning rate

Vectorization Approach

Linear Regression

Feature **Label**

area	price
6.7	9.1
4.6	5.9
3.5	4.6
5.5	6.7

House Price Data

$$x = \begin{bmatrix} 1 & 6.7 \\ 1 & 4.6 \\ 1 & 3.5 \\ 1 & 5.5 \end{bmatrix} \qquad y = \begin{bmatrix} 9.1 \\ 5.9 \\ 4.6 \\ 6.7 \end{bmatrix}$$

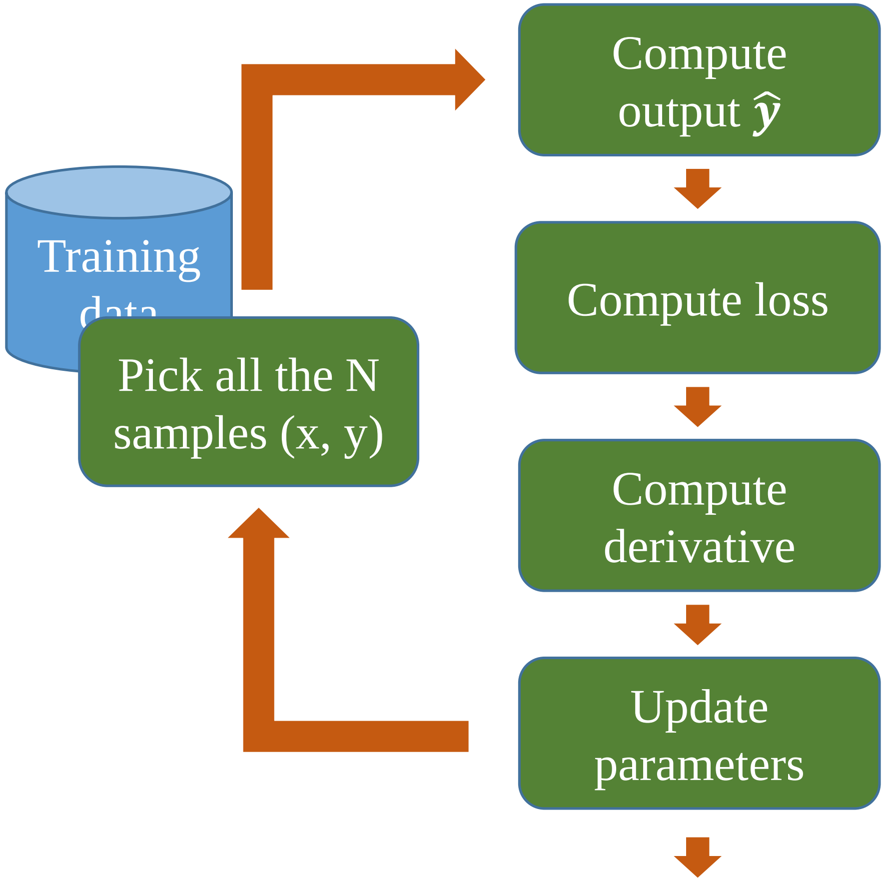
Model

$$\text{price} = \text{w} * area + b$$

$$\hat{y} = wx + b$$

Parameter Initialization

$$\boldsymbol{\theta} = \begin{bmatrix} b \\ w \end{bmatrix} = \begin{bmatrix} 0.049 \\ -0.34 \end{bmatrix}$$



1) Pick all the N samples from training data

2) Compute output $\hat{\boldsymbol{y}}$

$$\hat{\boldsymbol{y}} = \boldsymbol{x}\boldsymbol{\theta}$$

3) Compute loss

$$\boldsymbol{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = (\hat{\boldsymbol{y}} - \boldsymbol{y}) \odot (\hat{\boldsymbol{y}} - \boldsymbol{y})$$

4) Compute derivative

$$\boldsymbol{k} = 2(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

$$L'_{\boldsymbol{\theta}} = \boldsymbol{x}^T \boldsymbol{k}$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \frac{L'_{\boldsymbol{\theta}}}{N}$$

η is learning rate

```
for epoch in range(epoch_max):
    y_hat = x.dot(theta)

    loss = np.multiply((y_hat-y), (y_hat-y))
    losses.append(np.mean(loss))

    k = 2*(y_hat-y)
    gradients = x.T.dot(k) / N

    theta = theta - lr*gradients
```

Linear Regression

	Advantages	Disadvantages
1 sample	Simple to understand and implement Faster learning on some problems Noisy update is beneficial sometime	Computationally expensive Noisy gradient signal Convergence problem
m sample	A balance between the robustness of 1-sample and the efficiency of N-sample	
N sample	Computationally efficient More stable error gradient parallel processing	Premature convergence Memory problem Training speed is slower

Outline

- Review on One-sample Training
- Vectorize the Linear Regression (m-sample)
- Vectorize the Linear Regression (N-sample)
- Proofs of Some Matrix Properties

$A \in \mathcal{R}^{m\times n}$

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

$B \in \mathcal{R}^{m\times n}$

$$B = \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \dots & \dots & \dots \\ b_{m1} & \dots & b_{mn} \end{bmatrix}$$

Prove A + B = B + A

$$A + B = \begin{bmatrix} (a_{11} + b_{11}) & \dots & (a_{1n} + b_{1n}) \\ \dots & \dots & \dots \\ (a_{m1} + b_{m1}) & \dots & (a_{mn} + b_{mn}) \end{bmatrix}$$

$$= \begin{bmatrix} (b_{11} + a_{11}) & \dots & (b_{1n} + a_{1n}) \\ \dots & \dots & \dots \\ (b_{m1} + a_{m1}) & \dots & (b_{mn} + a_{mn}) \end{bmatrix} = B + A$$

$A \in \mathcal{R}^{m\times n}$

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

Prove $(A^T)^T = A$

$$(A^T)^T = \left(\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}^T\right)^T$$

$$= \left(\begin{bmatrix} a_{11} & \dots & a_{m1} \\ \dots & \dots & \dots \\ a_{1n} & \dots & a_{mn} \end{bmatrix}\right)^T$$

$$= \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} = A$$

$A \in \mathcal{R}^{m\times n}$ $\quad$ $I \in \mathcal{R}^{n\times n}$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Prove $IA = A = AI$

$$AI = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$= \begin{bmatrix} (a_{11}1 + a_{12}0+\dots+a_{1n}0) & (a_{11}0 + a_{12}1+\dots+a_{1n}0) & \dots & (a_{11}0 + a_{12}0+\dots+a_{1n}1) \\ (a_{21}1 + a_{22}0+\dots+a_{2n}0) & (a_{21}0 + a_{22}1+\dots+a_{2n}0) & \dots & (a_{21}0 + a_{22}0+\dots+a_{2n}1) \\ \vdots & \vdots & \vdots & \vdots \\ (a_{m1}1 + a_{m2}0+\dots+a_{mn}0) & (a_{m1}0 + a_{m2}1+\dots+a_{mn}0) & \dots & (a_{m1}0 + a_{m2}0+\dots+a_{mn}1) \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$